

EVENT TRACER

Patent Number: JP5204710
Publication date: 1993-08-13
Inventor(s): HASHIMOTO KAZUYA
Applicant(s):: NEC CORP
Requested Patent: ☐ JP5204710
Application JP19920034526 19920124
Priority Number(s):
IPC Classification: G06F11/28 ; G06F11/34
EC Classification:
Equivalents:

Abstract

PURPOSE: To realize an event trace function reducing influences exerted upon the performance or quality of a program itself subject to debugging.

CONSTITUTION: When a user does not activate the trace function, while referring to an event processing address 201, an event processing address recognizing means 101 executes a normal processing by moving execution control to an event processing program 301. When activating the trace function, an event processing address changing means 103 saves the contents of the event processing address 201 to an event processing address 202 before change, and the execution control is moved to an event trace information output means 102 at the time point of generating an event by setting an entry address to the event trace information output means 102. After trace information is outputted, an event processing address recognizing means 104 before change moves the execution control to the contents of the event processing address 202 before change so that event trace information can be collected.

Data supplied from the esp@cenet test database - I2

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-204710

(43) 公開日 平成5年(1993)8月13日

(51) Int.Cl.⁵

G 0 6 F 11/28
11/34

識別記号

3 2 0 A 9290-5B
L 9290-5B

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数3(全4頁)

(21) 出願番号 特願平4-34526

(22) 出願日 平成4年(1992)1月24日

(71) 出願人 000004237

日本電気株式会社
東京都港区芝五丁目7番1号

(72) 発明者 橋本 一也

東京都港区芝五丁目7番1号 日本電気株
式会社内

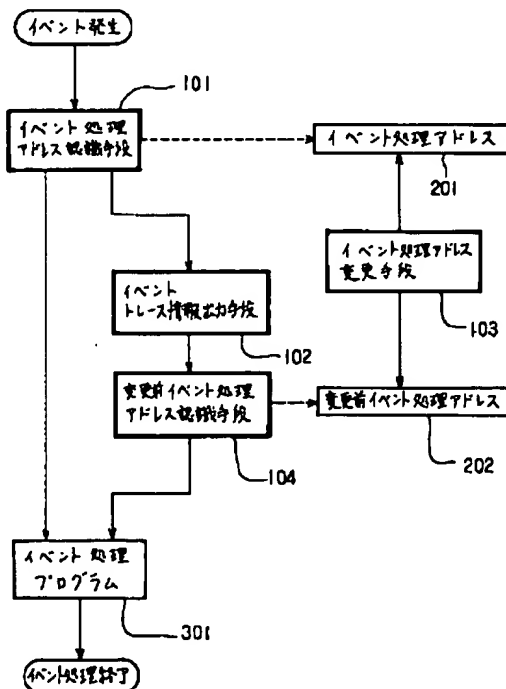
(74) 代理人 弁理士 井ノ口 壽

(54) 【発明の名称】 イベントトレース装置

(57) 【要約】

【目的】 デバッグ対象プログラム自体の性能や品質に与える影響が少ないイベントトレース機能を実現することにある。

【構成】 ユーザがトレース機能を活性化しないときには、イベント処理アドレス認識手段101がイベント処理アドレス201を参照し、イベント処理プログラム301に実行制御を移すことにより通常の処理を行う。トレース機能を活性化するときには、イベント処理アドレス変更手段103により、イベント処理アドレス201の内容を変更前イベント処理アドレス202に退避し、イベント処理アドレス201にイベントトレース情報出力手段102へのエントリアドレスを設定することにより、イベントが発生した時点でイベントトレース情報出力手段102に実行制御を移し、トレース情報を出力した後、変更前イベント処理アドレス認識手段104により、変更前イベント処理アドレス202の内容に実行制御を移すことによりイベントトレース情報を収集する。



1

【特許請求の範囲】

【請求項1】 イベントが発生した時点で、前記イベントに対するイベント処理プログラムのエントリアドレスを示すイベント処理アドレスを保持し、前記イベント処理アドレスを認識し、前記イベント処理アドレスに実行制御を移すためのイベント処理アドレス認識手段と、変更する以前のイベント処理アドレスを保存し、前記イベント処理アドレスをイベントトレース情報出力プログラムへのエントリアドレスに変更するためのイベント処理アドレス変更手段と、
前記イベントトレース情報出力プログラムを実行した後、前記イベント処理アドレス変更手段で保存された変更前イベント処理アドレスを認識し、
前記変更前イベント処理アドレスに実行制御を移すように制御するための変更前イベント処理アドレス認識手段と、
イベントが発生したときに情報を出力するためのイベントトレース情報出力手段とを備えたイベントトレース装置。

【請求項2】 前記イベントが発生したときには、前記イベント処理アドレス認識手段により前記イベント処理アドレスを参照し、前記イベント処理アドレスに格納された前記イベントトレース情報出力手段のエントリアドレスにより前記イベントトレース情報が出力されるように構成した請求項1のイベントトレース装置。

【請求項3】 トレース機能が活性化されていないときには前記イベント処理アドレス認識手段から直接的にイベント処理プログラムに実行制御を移し、前記トレース機能が活性化されているときには所定のトレース機能を実行してから前記イベント処理プログラムに実行制御を移すように構成し、かつ、前記トレース機能を活性化状態から不活性化状態に転ずるには前記変更前イベント処理アドレスの値をイベント処理アドレスの値に置換して行うように構成した請求項1のイベントトレース装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、プログラムのデバッグ方式に関し、特にそのイベントトレース機能に関する。

【0002】

【従来の技術】 従来、OSのシステムコール等のOSがサービスする機能を使用して動作するOS状のプログラムや、割込みなどの外部イベントに対して動作するプログラムを含めたシステム全体のプログラムの動作を確認する方法として、エミュレータ等の高価なハードウェアでシステムコールの発行履歴や外部イベントの発生状況などの実行トレースデータを収集し、そのデータをもとに解析をする方法が公知であった。ところが、この方法には、実行トレースデータの収集に量的な制約があり、また、実行トレースデータ中に余分な情報も収集されているため、その中から必要な情報を検索するのは困難で

2

あった。特に上述のようなシステム全体のプログラムの動作を確認するためには収集されるデータ量が十分ではなく、効率的ではなかった。

【0003】 一方、上述の収集機能をソフトウェアで実現するためには、OSのシステムコール発行や外部イベント等のイベントが発生した時点と、発生したイベントに対する終了時点とを捕らえ、次の方法によっていた。すなわち、第1にその時点に、トレースメッセージを表示するプログラムを埋め込むか、第2にデバッグシステムを利用し、その時点にブレーク機能を活性化し（ブレークポイントの設定）、該ブレークポイントでブレークした後にデバッグコマンドを利用することにより情報を表示する事で、上述の収集機能を実現していた。

【0004】 しかし、上述の方式には、以下の問題点がある。第1の方式では、デバッグが完了した時点で埋め込んだトレースメッセージ表示用のプログラムを取り外し、再度、コードを生成する必要があるため再検査も必要となる。また、トレースメッセージ表示用のプログラム分だけトレース対象プログラム中のメモリ容量を消費する。一方、第2の方式でひとつのトレース情報を得るためには、ブレークポイントの設定を行った後、実行コマンドにより実行再開、ブレークポイントにより実行停止および表示コマンドによりトレース情報表示を繰り返す手順でデバッグコマンドを実行する必要がある。従って、トレース情報を必要な時点から必要な分だけ得たい場合にはその必要な時点まで到達するのに時間がかかったり、あるいはデバッグコマンドの起動回数が多くなる。また、第2の方式では、トレース情報を収集するための処理時間が大きく、トレース機能を使用しているときの実行速度はトレース機能を使用していないときの実行速度と比べ大きい。

【0005】

【解決しようとする課題】 解決しようとする問題点は、第1の方法で一つのトレース情報を得るためには、ブレークポイントの設定を行った後、複数の手順を繰り返す必要があるため、処理時間が長く、かつデバッグコマンドの起動回数が多くなるという点であり、第2の方法では処理時間が長く、かつ、実行速度が低下する点である。

【0006】

【課題を解決するための手段】 本発明のイベントトレース装置は、デバッグ機能としてデバッグ対象プログラムのイベント発生状況をトレースする機能を具備するデバッグシステムにおいて、イベントが発生した時点で、イベント処理アドレス認識手段は該イベントに対するイベント処理プログラムのエントリアドレスを示すイベント処理アドレスを保持し、該イベント処理アドレスを認識し、該イベント処理アドレスに実行制御を移し、イベント処理アドレス変更手段は変更する以前のイベント処理アドレスを保存し、該イベント処理アドレスをイベント

トレース情報出力プログラムへのエントリアドレスに変更し、変更前イベント処理アドレス認識手段はイベントトレース情報出力プログラムを実行した後、該イベント処理アドレス変更手段で保存された変更前イベント処理アドレスを認識し、該変更前イベント処理アドレスに実行制御を移すことを制御することを特徴とする。

【0007】

【実施例】次に、本発明について図面を参照して説明する。図1は、本発明によるイベントトレース装置の一実施例を示すブロック図である。図1において、101はイベント処理アドレス認識手段、102はイベント情報出力手段、103はイベント処理アドレス変更手段、104は変更前イベント処理アドレス認識手段、201はイベント処理アドレス、202は変更前イベント処理アドレス、301はイベント処理プログラムである。

【0008】図1において、イベント処理認識アドレス認識手段101は、イベント処理アドレス201を認識し、該イベント処理アドレス201へ実行制御を移す。イベント情報出力手段102は、イベントが発生したという情報を出力する。イベント処理アドレス変更手段103は、イベント処理アドレス201の内容を変更前イベント処理アドレス202に退避し、イベント処理アドレス認識手段101が参照して、実行制御を移すエントリアドレスであるイベント処理アドレス201の内容を変更する。イベント処理アドレス認識手段104は、変更前イベント処理アドレス202の内容を参照し、該アドレスへ制御を移す。イベント処理プログラム301は、イベントに対する処理プログラムである。

【0009】以下に第1図をもとに本発明の動作を詳細に説明する。本実施例においては、イベントが発生した際に、イベントが発生した事をトレースデータとして収集することとする。まず、トレース機能を活性化しない場合の処理を説明する。イベントが発生したならば、イベント処理アドレス認識手段101によりイベント処理アドレス201を参照し、イベント処理301に実行制御を移す。次に、トレース機能を活性化する場合には、イベント処理アドレス変更手段103により、イベント処理アドレス201の内容を変更前イベント処理アドレス202に退避して、イベントトレース情報出力手段102のエントリアドレス値をイベント処理アドレス201の内容に変更する。

【0010】イベントが発生した場合、イベント処理アドレス認識手段101によりイベント処理アドレス201を参照する。イベント処理アドレス201には、イベントトレース情報出力手段102へのエントリアドレス

が格納されているので、イベントトレース情報出力手段102へ実行制御が移り、イベントトレース情報が出力される。さらに、変更前イベント処理アドレス認識手段104は、イベント処理アドレス変更手段103により退避された変更前イベント処理アドレス202を参照し実行制御を移す。変更前イベント処理アドレス202には、イベント処理301へのエントリアドレスが格納されているので、変更前イベント処理アドレス認識手段104によりイベント処理301が実行される。

【0011】トレース機能を活性化している状態から活性化しないようにするためには、イベント処理アドレス変更手段103により、変更前イベント処理アドレス202の内容を、イベント処理アドレス201の内容とすることにより達成することができ、容易にトレース機能の制御を行うことができる。

【0012】

【発明の効果】以上説明したように本発明は、比較的簡単な操作により、デバッグ対象プログラム自体の性能や品質面に大きく影響を与えることなくイベントのトレース機能を提供することができ、システム全体のデバッグ効率を高めることができる利点がある。また、被デバッグ対象プログラムとトレース機能とは独立に存在させることができるため、メモリ容量にも負担をかけず、容易にトレース機能を取り外すこともできる利点がある。イベント処理アドレス認識手段101を使用し、イベント処理アドレス201を定義して、イベント処理プログラム301をトレースする対象のプログラム中に存在させることができる。また、イベント情報出力手段102、変更前イベント処理アドレス認識手段104およびイベント処理アドレス変更手段103を使用し、変更前イベント処理アドレス202を定義してデバッガ内のトレース機能を実現すると、デバッグ対象プログラム自体のプログラム容量や、トレース機能を使用しないときの処理性能に影響が少ないという利点がある。

【図面の簡単な説明】

【図1】本発明によるイベントトレース装置の一実施例を示すブロック図である。

【符号の説明】

- 101 イベント処理アドレス認識手段
- 102 イベント情報出力手段
- 103 イベント処理アドレス変更手段
- 104 変更前イベント処理アドレス認識手段
- 201 イベント処理アドレス
- 202 変更前イベント処理アドレス
- 301 イベント処理プログラム

【図1】

